Journal of Logical and Algebraic Methods in Programming 102 (2019) 46–63

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Stubborn versus structural reductions for Petri nets

Frederik M. Bønneland *, Jakob Dyhr, Peter G. Jensen *, Mads Johannsen, Jiří Srba *

Department of Computer Science, Aalborg University, Selma Lagerlöfs Vej 300, DK-9220 Aalborg East, Denmark

A R T I C L E I N F O

Article history:
Received 12 March 2018
Received in revised form 6 July 2018
Accepted 7 September 2018
Available online 14 September 2018

Keywords:
Partial order reduction
Stubborn sets
Structural reductions
Reachability analysis
Petri nets

A B S T R A C T

Partial order and structural reduction techniques are some of the most beneficial methods for state space reduction in reachability analysis of Petri nets. This is among others documented by the fact that these techniques are used by the leading tools in the annual Model Checking Contest (MCC) of Petri net tools. We suggest improved versions of a partial order reduction based on stubborn sets and of a structural reduction with additional new reduction rules, and we extend both methods for the application on Petri nets with weighted arcs and weighted inhibitor arcs. All algorithms are implemented in the open-source verification tool TAPAAL and evaluated on a large benchmark of Petri net models from MCC'17, including a comparison with the tool LoLA (the last year winner of the competition). The experiments document that both methods provide significant state space reductions and, even more importantly, that their combination is indeed beneficial as a further nontrivial state space reduction can be achieved.

1. Introduction

Model checking of large distributed and concurrent systems is often limited in its applicability due to the state space explosion problem. Components in concurrent systems may independently perform actions without being in conflict with other components, forcing an explicit state space analysis to explore every possible interleaving of the actions and hence creating an explosion in the number of executable action sequences. Petri nets are a popular formalism for modeling of concurrent systems [12], however, due to the state space explosion problem, essentially all interesting questions about their behavior, including the reachability and coverability problems, are EXPSPACE-hard (see e.g. [4]).

Despite the discouraging complexity results, numerous techniques have been developed to improve the feasibility of reachability analysis, including methods based on reducing the state space by eliminating the interleaving in independent components (see e.g. [5,1]). The focus of our work is on two such techniques: structural reductions [11] and stubborn set reductions [15], both applied to and evaluated on the model of weighted Petri nets with inhibitor arcs. *Structural reductions* preprocess the Petri net model by collapsing redundant places and transitions, while preserving the validity of the model checking question. The idea is that a smaller number of places and transitions in a net can help to reduce the degree of concurrency and eliminate some unnecessary interleavings. In partial order reductions, like e.g. *stubborn set reduction*, we identify transitions that are independent of each other and the order of their execution does not influence the model checking property in question. This can be considered as another method that can, in an on-the-fly manner, reduce the number of possible interleavings of independent actions. Both structural and stubborn reductions can be in a straightforward

* Corresponding authors.
E-mail addresses: frederikb@cs.aau.dk (F.M. Bønneland), pgj@cs.aau.dk (P.G. Jensen), srba@cs.aau.dk (J. Srba).

https://doi.org/10.1016/j.jlamp.2018.09.002
2352-2208/

way combined, however, to the best of our knowledge, the effect of this combination has not previously been studied in detail.

We perform a comparative study of the effects of the two types of state space reduction techniques and their combination. For our experiments, we use the database of nets and reachability queries from the annual Model Checking Contest (MCC) [8] and conclude that while both techniques are clearly beneficial for the performance of the reachability analysis, the combination of the two methods demonstrates yet another degree of performance improvements. Apart from this experimental evaluation, we make several technical contributions to stubborn and structural reductions applied to the model of Petri nets. Both techniques are extended to work for the reachability logic used in MCC, while allowing us to use weighted arcs as well as weighted inhibitor arcs. In particular, the stubborn set reduction as well as the structural reduction were refined to take weighted arcs into account, in order to minimize the size of the state space that is necessary to explore for a given reachability query. In stubborn reduction, we refine the computation of dependencies between transitions so that instead of the traditional comparison of presets and postsets of places, we utilize a more detailed analysis of the increasing/decreasing effect of a transition on a given place. All techniques are proved correct and implemented in the model checker TAPAAL [3]. The experiments are encouraging as the improved techniques in their combination allow us to solve more reachability queries from MCC'17 [8] than the model checker LoLA [21], the last year winner in the reachability category.

Related work. The stubborn reduction technique is related to and based upon the seminal work on stubborn sets by Valmari et al. [18,13,9,15,16]. This includes write up/down sets [15], the closure procedure [9], and attractor sets [13]. We contribute by adding support for inhibitor arcs, extending the technique to a reachability logic used in MCC and presenting a different formulation of stubborn sets for reachability in the general setting of labeled transition systems. Further analysis during the generation of stubborn sets can help to generate more optimal (smaller) stubborn sets, which can be done e.g. by extracting terminal strongly connected components from the derived transition dependency graph [18]. We choose to use instead heuristic methods for the generation of stubborn sets as they have smaller computational overhead and achieve better performance in our experiments.

Structural reductions of Petri nets were studied by Murata et al. [11,10] with the main focus on preserving liveness, safety, and boundedness. The reduction rules were recently extended to include weighted nets with inhibitor arcs while preserving the reachability of cardinality queries [6]. We contribute by increasing the applicability of the four rules presented in [6] and refining them for the use with weighted arcs so that a more significant net reduction can be achieved compared to [6]. Moreover, we introduce five new reduction rules, allowing us to reduce the size of the input net even further.

Stubborn sets are also an important state space reduction technique used in the tool LoLA [21] that we compare against to in our experiments. Their stubborn set implementation has several approaches to reachability analysis, utilizing up/down sets and terminal strongly connected components [9] to mention some. Approaches using terminal strongly connected components can present some performance problems due to concurrent cycles of invisible (or non-interesting) transitions, forcing the method to sometimes explore the full parallel composition [17,19]. Remedies to this have been explored in the form of frozen actions [17], removing transitions from consideration if they are tagged as frozen. Besides LoLA's take on stubborn sets [13], their tool includes several other reduction and verification improvements such as symmetry reduction [14] and Counter Example Guided Abstraction Refinement (CEGAR) [20], however, LoLA does not employ structural reductions. Our experiments document that the refined and combined application of our stubborn and structural reduction techniques becomes competitive in performance compared with the tool LoLA.

2. Preliminaries

A *labeled transition system* (LTS) is a tuple $TS = (\mathcal{S}, A, \rightarrow)$ where $\mathcal{S}$ is a set of states, A is a set of actions (or labels), and $\rightarrow \subseteq \mathcal{S} \times A \times \mathcal{S}$ is a transition relation. We write $s \xrightarrow{a} s'$ whenever $(s, a, s') \in \rightarrow$ and say that a is *enabled* in s. The set of all enabled actions in a state s is denoted $en(s)$. A state s is a *deadlock* if $en(s) = \emptyset$. We write $s \rightarrow s'$ whenever there is an action a such that $s \xrightarrow{a} s'$. We inductively extend the relation $\xrightarrow{a}$ to sequences of transitions $w \in A^*$ such that $s \xrightarrow{\epsilon} s$ and $s \xrightarrow{wa} s'$ if $s \xrightarrow{w} s''$ and $s'' \xrightarrow{a} s'$. We write $s \rightarrow^n s'$ if there is $w \in T^*$ of length n such that $s \xrightarrow{w} s'$, and we write $s \rightarrow^* s'$ if $s \rightarrow^n s'$ for some $n \geq 0$.

The *reachability problem* is, given an LTS $TS = (\mathcal{S}, A, \rightarrow)$, an initial state $s \in \mathcal{S}$, and a set of goal states $G \subseteq \mathcal{S}$, to decide whether there is $s' \in G$ s.t. $s \rightarrow^* s'$.

2.1. Petri nets

Let $\mathbb{N}^0 = \mathbb{N} \cup \{0\}$ be the set of natural numbers including 0. Let $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$ be the set of natural numbers including infinity.

Definition 1 *(Petri net with inhibitor arcs).* A Petri net is a tuple $N = (P, T, W, I)$ where P and T are finite and disjoint sets of places and transitions, $W\colon (P \times T) \cup (T \times P) \rightarrow \mathbb{N}^0$ is a weight function for regular arcs, and $I : (P \times T) \rightarrow \mathbb{N}^\infty$ is a weight function for inhibitor arcs.